

DOE Workshop on Science Collaborations for Extreme Scale Science



Jeff Kantor

Large Synoptic Survey Telescope
Data Management Systems Manager

LSST Data Management System

- Every ~ 17 seconds, data is read out of 3.2 Gpixel camera in 2 seconds, transported 70 km over fibers in Chile
- Process for detection of transients against all known sources in the sky (billions) within 60 seconds
- Transfer raw data to Illinois for archiving
- Re-process all accumulated data from start of survey every year (70 PB in year 10)
- Serve data to worldwide community of scientists
- Enable data access by general public, primary and secondary students/educators (serveEPO system)

Challenges for LSST and ground-based optical/infra-red astronomy

- Data volumes (~15 TB raw data/night) to process
- High reliability, inter-continental data transfer
- Maintain complete data integrity for decades, no data lost
- Parallel, fast computation (~1.6 PetaFLOPS yr 10)
- Parallel, fast, flexible query/data access to a petascale database (~425 GB/s aggregate)
- Order of 10^3 simultaneous, geographically distributed users, virtual organizations
- Much of the interesting science is in the very low S/N area where systematic errors can be deadly
- Need cross-survey fusion (e.g. optical w/spectroscopy, x-ray, radio)
- Observational/empirical and theoretical interplay (we don't always know what we are looking for)

Priority Research Direction: Programming in hybrid CPU/GPU environments

Key emerging challenges

- Parallelization at 500x smaller RAM/core ratios than CPU systems
- Immaturity and multiplicity of GPU programming models and APIs
- Poor debugging in hybrid environments
- Inaction will lead to high cost, underuse, fragmentation, and/or vendor dependence

Potential impact on software/systems

- Increased ease of programming in hybrid environments
- Unification of CPU/GPU and CPU-only massively parallel programming paradigms
- Independence on hardware platforms
- “Future-proofing” of massively parallel code

Summary of research direction

- Algorithm parallelization for GPUs
- Single programming paradigm/library/toolchain that can target to both GPUs and CPUs
- Profiling tools to help determine optimal mix, allocation of processing

Potential impact on usability, capability, and breadth of community

- Lowering overhead and barriers to programming at large scale
- Improved “bang for the computing buck” offered by GPUs realized more widely
- Commoditization of exascale systems, emergence of “everyday” exascale science

Priority Research Direction: Distributed/parallel query of extreme scale databases

Key emerging challenges

- Transparent failure recovery
- Reducing inter-node traffic
- Debugging distributed systems
- Result aggregation
- Distributed data integrity
- Reducing I/O

Potential impact on software/systems

- Can rely on cheap commodity
- Easy incremental scaling
- Complexity in software (fault tolerance)

Summary of research direction

- Fault tolerance in software
- Data-aware partitioning, overlaps, multi-level
- 2-level aggregation (map/reduce style)
- Shared scans

Potential impact on usability, capability, and breadth of community

- Can enable new science, previously too expensive
- Can enable better QA, previously too expensive

Priority Research Direction: Curation of extreme scale datasets for decades

Key emerging challenges

- Longevity = major migrations unavoidable (software, OS, hardware/technology)
- Funding: individual projects/institutions are not geared to long-term curation
- Virtualization/federation can help to some degree

Potential impact on software/systems

- Software more complex, virtual
- Expensive migrations vs. portable data

Summary of research direction

- Virtual machines, communities
- Federated databases
- Dividing software into components, layering, shielding
- Narrow interfaces

Potential impact on usability, capability, and breadth of community

- Can enable more science by making data available as long as valuable, e.g. longer time series